ESTIMATION OF SPEAKER COUNT USING FEATURES LEARNT FROM SUPERVISED AND UNSUPERVISED DEEP LEARNING METHODS

Ashwin Kalyan V, Gururaj Krishnamurthy, Vijayakrishna Naganoor, Deepu Vijayasenan

Music and Audio Research Group
Department of Electronics and Communication Engineering
National Institute of Technology Karnataka

ABSTRACT

Determining the number of speakers participating in a conversation without any prior information about the speakers is a challenging problem, especially when the audio samples are short. Solving it can improve the performance of speaker diarisation systems. This paper aims at identifying the number of speakers in an unsupervised manner by employing features learned from deep learning methods. Going beyond the conventional MFCC features, this paper explores the usage of features extracted from Deep Convolutional Neural Networks (DCNN) and Convolutional Deep Belief Networks (CDBN) to solve the problem at hand.

***Index Terms*—** speaker count, diarisation, CDBN, DCNN

1. INTRODUCTION

Identifying the number of speakers in a conversation is crucial for speech related tasks including recognition. For example, this can be used to enhance the performance of Automatic Speech Recognition (ASR) systems by enabling speaker adaptation. In addition, information about the duration of each speaker's activity could be utilized in social signal processing tasks such as dominant speaker identification.

Researchers have relied on microphone array recordings [1] to solve this problem for scenarios like meeting data. However, similar efforts that use single-channel recordings are not very prominent. Most of the state of the art algorithms that perform the speaker diarisation task follow a clustering based approach [2]. An agglomerative clustering is performed on short speech segments using a Bayesian Information Criterion (BIC). The algorithm simultaneously determines the number of speakers as well as the speech segments of each speaker. [3, 4]. The agglomerative clustering is initialized with some speech segmentation algorithm (often uniform linear segmentation) to come up with an overdetermined number of speakers.

However in the case of short speech recordings, it is difficult to adopt the above mentioned approach primarily because of availability of lesser data to detect similar speech clusters and train individual speaker models. Another plausible reason is the use of conventional features like Mel-Frequency Cepstral Coefficients (MFCC) and Linear Predictive Cepstral Coefficients (LPCC) that are not tailored to represent speaker-dependent information.

An alternative scheme [5] that is based on modelling the distribution of inter and intra speaker distances is effective for short segments as it has a relatively lesser dependence on the duration of the audio. The mentioned work makes use of LPCC features and uses mahalanobis distance between speech segments to model the inter and intra speaker distance distributions. For such methods, using features that have very good speaker representations can result in higher accuracies.

In this work, methods to improvise the above mentioned approach using better feature representations and distance metrics are explored. More specifically, the effectiveness of neural network based methods to learn better feature representations has been analysed. Deep Convolutional Neural Networks (DCNN) have seen unprecedented success in the field of computer vision for tasks like object recognition. Middle level features extracted from DCNNs like Alexnet [6] have been fine-tuned to suit various image classification and recognition tasks with reasonable performance. Although many interesting works on tasks like speech recognition [7] and speaker recognition [8] are present, the effectiveness of deep neural networks in the field of audio is still not well researched. To explore the applicability of this supervised technique for the speaker count problem, we train a DCNN to perform speaker classification and use the middle-level features from the net to compute the speaker distance distributions.

Motivated by [9], we also explore the use of features learned through unsupervised learning in this work. Convolutional Deep Belief Networks (CDBN) are trained on speech spectra to obtain the unsupervised higher level representation. This paper proves the effectiveness of CDBN features on standard tasks like speaker verification, phoneme recognition and gender recognition and obtains comparable or better performance against the use of conventional MFCC features.

In this work, we try to exploit the higher level representation achieved by the CDBN features to compute the inter and intra speaker distance distributions.

2. GENERALIZED RESIDUAL RATIO ALGORITHM

Before proceeding to the details of our work, we briefly explain the method used in [5] as it forms the basis for the improvisations made in this paper.

Speaker models are formed by grouping consecutive voiced segments from the same utterance and speaker. Then, intra and inter speaker distances between the models are computed using the mahalanobis distance. This metric performs better than metrics like euclidean distance as it takes into account the correlations of the dataset. The length of the speech model is crucial as choosing smaller or larger lengths results in poor separation between the inter and intra distance distributions. The speech models used are typically around 1 second.

Let the mean and variance of the intra-speaker and inter-speaker distances be μ_1, σ_1 and μ_2, σ_2 respectively. The gaussian probability of the distance x between two speech models can be computed using

$$f(x|\mu_i, \sigma_i) \quad for \quad i = 1, 2$$

where $f(x|\mu, \sigma)$ is the gaussian function. Once the distance distributions are computed using multiple speech models, the Maximum Likelihood Ratio (MLR) is used to determine if the mahalanobis distance between two speech models belong to the same speaker or to different speakers. MLR is defined as

$$MLR = \frac{f(x|\mu_1, \sigma_1)}{f(x|\mu_2, \sigma_2)} \tag{1}$$

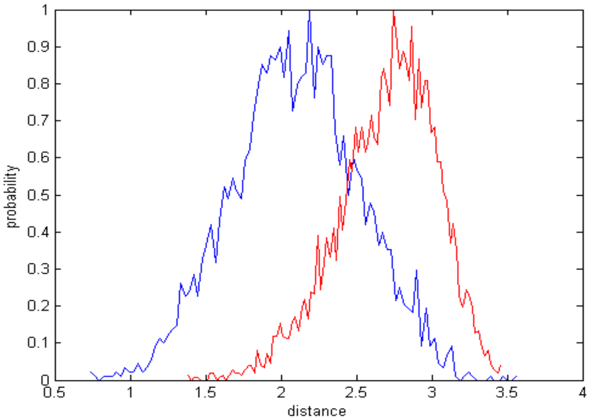

Fig. 1. Intra (red) and inter (blue) speaker distance distributions computed using MFCC features and mahalanobis distance measure

The Generalized Residual Ratio Algorithm (GRRA) is used to compute the number of speakers. A maximum of K speakers can be identified using this algorithm where K is set before running the algorithm. It consists of the following steps:

1. Speech models are formed from a given conversation.
2. All pairwise distance between all models are computed
3. A reference model is chosen at random and MLR tests are performed between this model and all others. Every model with $MLR > \lambda$ (if inter and intra speaker distances are assumed to be equally probable, $\lambda = 1$) is considered to belong to the reference speaker, and is eliminated from the conversation along with the reference model itself. The Residual Ratio i.e. the ratio of the size of residual speech to the original size of the conversation is determined. This completes the first elimination round.
4. Step 3 is repeated for the second round. However, care must be taken in order to ensure that the newly chosen reference model is not one of those that belong to the first reference speaker but were erroneously mismatched in the first round. This is done by checking that the ratio of size of the speech that was matched to the second reference to the total amount of speech is greater than a previously determined threshold. The process is repeated until this ratio is greater than the threshold. Once this condition is satisfied, the Residual Ratio for the second round is determined.
5. Step 4 is repeated K-1 times.

Ideally, the K reference models chosen should belong to K different speakers (if K different speakers are present) and the residual ratio at the end should be 0. To determine the number of speakers from the K residual ratios, we use the Stopped Residual Ratio (SRR) method mentioned in the work.

In our work, we use all frames from the speech unlike the referred work where only voiced frames are used. Retaining the framework, we improvise upon this by using features extracted from neural networks. As the features tend to be of higher dimensionality compared to MFCC/LPCC features, other methods to compute distances between speech models are used. Using the mahalanobis distance is not suited when the number of instances is smaller than the dimension of the feature.

Throughout this work, we have used the TIMIT database [10] to generate the speech distance distributions. 8 sentences, each of about 3s from 200 speakers form the training data. All the accuracies reported in this paper are computed using the distance distributions from this set of speakers unless mentioned otherwise. The remaining speakers comprise the test set. Conversations with multiple speakers were created by concatenation of utterances from different speakers and each sample lasted roughly 30s.

3. DCNN FEATURES

In this section, the process of obtaining higher level features containing speaker dependent information using DCNN is detailed.

3.1. Data Preparation

MFCC features are extracted from the audio using a window size of 20ms and a hop size of 10ms. A hamming window is applied on the data before computing the spectrogram. The filter banks are applied for a frequency range of 0-8kHz and 13 cepstral coefficients are extracted. MFCCs from 80 frames (roughly 0.8s) are grouped to form a speaker model akin to the method described in the previous section. A DCNN is trained to perform a 200-way speaker classification. Two dimensional speech models of size 13×80 form the input to the net. The input is mean-centered by subtracting the mean speech model.

3.2. Architecture

The DCNN consists of 7 learned layers with 4 convolutional layers and 3 fully connected layers. All the convolutional layers use a kernel of size 5×5, stride of 1 and pad of 2. This preserves the dimensions of the data after each convolution. After each convolution layer, max-pooling is performed to downsample the data. The pooling kernel size is 1×2 so that after each pooling the number of rows in the data is halved. For example, after the first pooling layer, the data size is reduced to 13×40. 3 fully connected layers follow after the convolutional layers. The first two fully-connected layers are followed by dropout layers (with drop-out ratio 0.5) [11] to prevent overfitting. Rectified Linear Units (ReLU) activation units are used throughout the net due to their proven superiority over other activation functions like the sigmoid function. The softmax function is used to compute the loss of the network. An overview of the network architecture is pictorially presented in figure 2.

3.3. Details of Learning

We used Stochastic Gradient Descent with a batch size of 128 samples, momentum of 0.9 and weight decay of 0.005 to train the DCNN. We initialize the weights using the xavier [12] method and biases with zeros. The initial learning rate is set to 0.005 and the network was trained roughly 110 times over the complete training set. We used Caffe [13] to train our neural networks using GPU acceleration.

3.4. Accuracy of Learning

Using dropout strategy significantly improves the performance of the net as the loss on the testing set improves from 1.92 to 0.54. The training loss with dropout is 0.26. The net achieves a final accuracy of 96.4% on the training set and 84.3% on the testing set. It should be noted that this is the classification accuracy for the speech models and not for utterances. Employing this net to compute accuracies for whole utterances by using majority voting on the classification of constituent speech models has a performance comparable with the results mentioned in [9]

4. CDBN FEATURES

In this section, we explore the use of features obtained from unsupervised learning using CDBN.

4.1. Outline of feature extraction

We follow [9] for both data preparation and training of the CDBN. The spectrogram had a 20 ms window size with 10 ms overlaps. The spectrogram was further processed using PCA whitening (with 80 components) to reduce the dimensionality. We then trained 300 first-layer bases with a filter length of 6 and a max-pooling ratio (local neighbourhood size) of 3. The CDBN was trained using contrastive divergence. In this work, only single layer features (we will use CDBN-L1 to indicate these) of dimensionality 300 are used as these exhibit better performance in the case of speaker identification [9]. Once the weights are trained, the CDBN-L1 features are extracted for speech models to form a two dimensional representation of size 300×80. A shallow network with 2 hidden layers is trained on this data to perform a 200-way speaker classification. The features from the second layer of the network (of dimensionality 200) are used to compute the intra and inter speaker distance distributions as explained previously in the context of DCNN features. Stochastic gradient descent with a batch size of 128, momentum of 0.9 and a weight decay of 0.005 was used to perform the training. The learning rate was initialized to 0.03. Dropouts (dropout ration of 0.4) were used to reduce overfitting.

4.2. Accuracy of Learning

The net achieved a loss of 0.84 on the training set and a loss of 1.56 on the testing set. Similar to DCNN, using dropouts in this shallow network prevented overfitting. The network was trained roughly 140 times on the complete training set.

5. RESULTS

The accuracy of our method in determining speaker count was evaluated by concatenating speech from different speakers randomly from the TIMIT database. The accuracies are

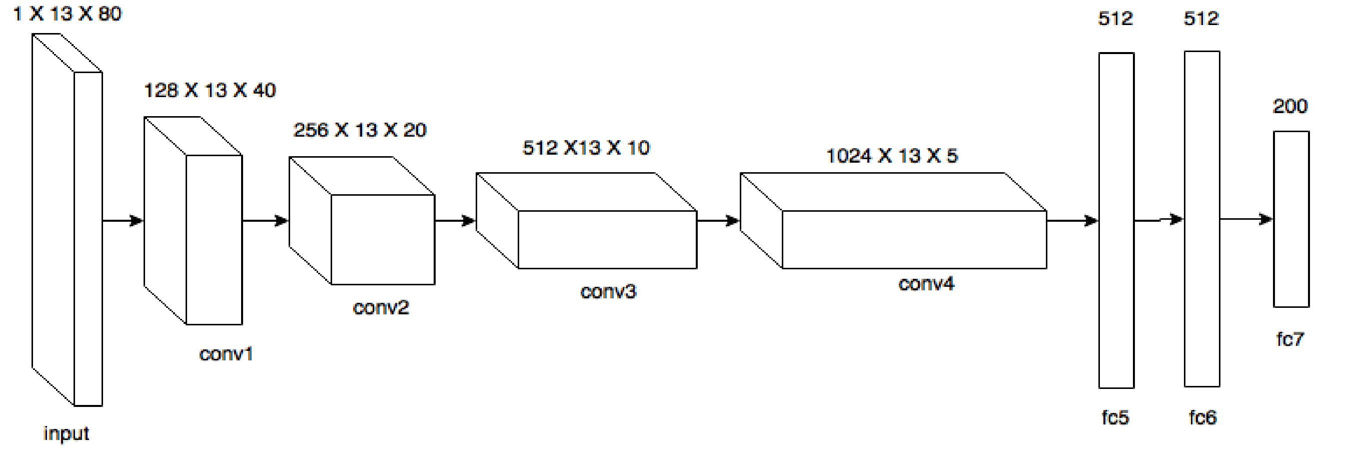

Fig. 2. Overview of the DCNN architecture. Every learned layer uses a ReLU activation function and each of the convolutional layers indicated is followed by a max pooling layer that is not indicated in the above figure. The loss is computed from the output of the fc7 layer.

reported for cases where speakers in the audio are part of the training set and for cases where from speakers are not present in the training set.

Case	MFCC	DCNN	CDBN
1 or more	83.3	98.8/ 85.3	96.5/ 90.4
2 or more	64.6	97.6/ 65.5	94.4/ 72.6
3 or more	53.2	95.8/ 61.1	92.1/ 68.7

Table 1. Percentage accuracies for speaker-count problem using the different features proposed in this paper and using the baseline model. Two accuracies are given for the DCNN and the CDBN features where the first value corresponds to speakers in the training data and the second value corresponds to speakers not part of the training data.

5.1. DCNN features

The 200 dimensional feature vectors from the final layer of the network which are now a higher level representation of the input speech models are used to compute intra and inter speaker distance distributions.Cosine distance is used to compute the distance between two feature vectors. The distance distributions using speakers present in the training are given in figure 3 and the same for speakers not in the training set is given by figure 4. For the speakers present in the training set, the separability between the two distributions is very high while, the same is relatively lesser when speakers not in the training set are used. However, this separability is still better than the separability with conventional MFCC features as indicated by the increase in speaker count accuracy.

5.2. CDBN features

The final accuracy on the test set was 77.68%. Again, it should be noted that this accuracy is for speech models and not for whole utterances. The distance distributions resulting from the 200-dimensional feature learnt from the shallow net are shown in figures 5 and 6. Figure 5 shows the distance distributions using speakers present in the training set and figure shows the same for speakers not present in the training set. The CDBN features fare better than the DCNN features due to the increased separability of the two distributions.

6. CONCLUSION AND FUTURE WORK

Arriving at the feature that distinguishes speaker to speaker variations while accounting for factors like language, accent, etc. will help solve the problem of identifying the number of speakers in an unsupervised manner. We believe that our work is a preliminary effort in this direction. From our experiments it is evident that features learnt in an unsupervised manner are superior to features learnt in a supervised manner. Also, both the features (CDBN-L1 and DCNN features) seem to perform better as compared to conventional MFCC features. We believe that continuing similar work can result in a standardized method of producing features that are suited for a given audio or speech task, clocking higher performance than the usual MFCC features.

Regarding the specific task concerning this paper, we believe that carrying out these experiments on larger datasets with more variation in terms of language, accent, recording quality, background environment, etc. can produce a more general speaker-dependent feature that sees through variations due to channel effects, session effects, noise and

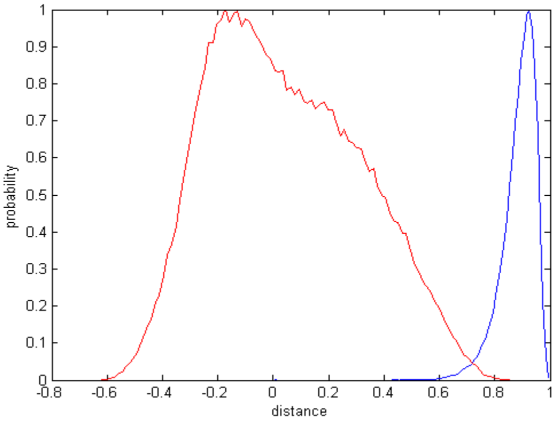

Fig. 3. Intra (red) and inter (blue) speaker distance distributions computed using DCNN features and cosine distance measure for speakers present in the training data.

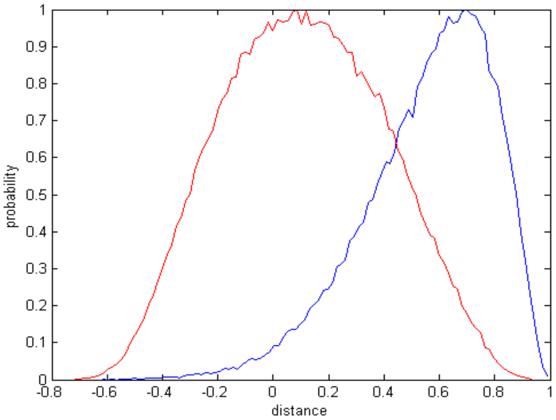

Fig. 4. Intra (red) and inter (blue) speaker distance distributions computed using DCNN features and cosine distance measure for speakers not present in the training data.

background. Also, with suitable post-processing this method can be used to improve the performance of tasks like speaker diarisation and segmentation.

7. REFERENCES

[1] Erich Zwyssig, Steve Renals, and Mike Lincoln, "Determining the number of speakers in a meeting using microphone array features," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4765–4768.

[2] Chuck Wooters and Marijn Huijbregts, "The icsi rt07s speaker diarization system," in *Multimodal Technologies for Perception of Humans*, pp. 509–519. Springer, 2008.

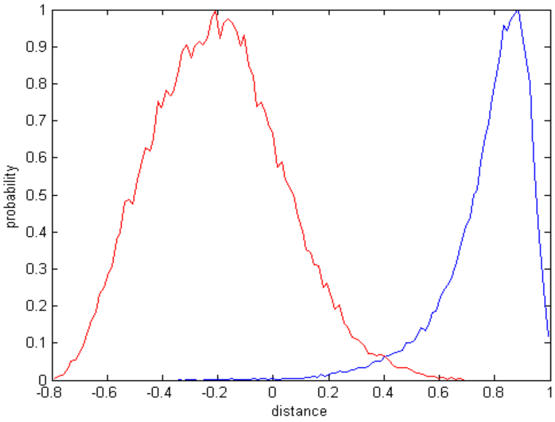

Fig. 5. Intra (red) and inter (blue) speaker distance distributions computed using CDBN features and cosine distance measure for speakers present in the training data.

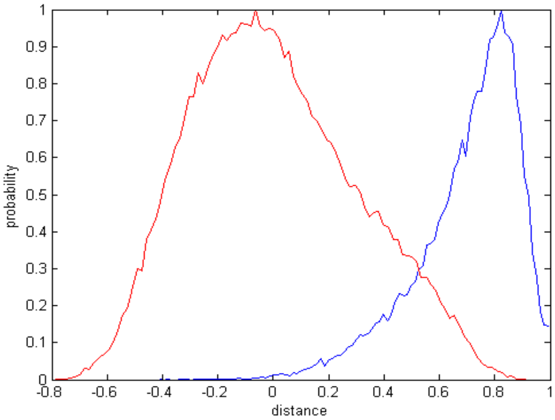

Fig. 6. Intra (red) and inter (blue) speaker distance distributions computed using CDBN features and cosine distance measure for speakers not present in the training data.

[3] Jitendra Ajmera, Guillaume Lathoud, and Iain McCowan, "Clustering and segmenting speakers and their locations in meetings," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*. IEEE, 2004, vol. 1, pp. I–605.

[4] Jitendra Ajmera, Hervé Bourlard, Itshak Lapidot, and Iain A McCowan, "Unknown-multiple speaker clustering using hmm," Tech. Rep., IDIAP, 2002.

[5] Uchechukwu O Ofoegbu, Ananth N Iyer, Robert E Yantorno, and Brett Y Smolenski, "A speaker count system for telephone conversations," in *Intelligent Signal Processing and Communications, 2006. ISPACS'06. International Symposium on*. IEEE, 2006, pp. 331–334.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[7] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[8] Fred Richardson, Douglas Reynolds, and Najim Dehak, “Deep neural network approaches to speaker and language recognition,” 2015.

[9] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng, “Unsupervised feature learning for audio classification using convolutional deep belief networks,” in *Advances in neural information processing systems*, 2009, pp. 1096–1104.

[10] William M Fisher, George R Doddington, and Kathleen M Goudie-Marshall, “The darpa speech recognition research database: specifications and status,” in *Proc. DARPA Workshop on speech recognition*, 1986, pp. 93–99.

[11] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv preprint arXiv:1207.0580*, 2012.

[12] Xavier Glorot and Yoshua Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *International conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[13] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, “Caffe: Convolutional architecture for fast feature embedding,” *arXiv preprint arXiv:1408.5093*, 2014.